# Day 5: Supply Stacks

(besedilo naloge)

Imamo sliko skladišča in navodila, kaj prelagamo kam. Razlika med prvim in drugim delom je v tem, da v prvem delu navodilo "prestavi 3 pakete iz 2 na 4" izvedemo tako, da prestavimo te tri pakete enega za drugim (in tako obrnemo njihov vrstni red), v drugem delu pa prestavimo vse tri (ne da bi obrnili vrstni red).

```
    [D]
[N] [C]
[Z] [M] [P]
 1   2   3

move 1 from 2 to 1
move 3 from 1 to 3
move 2 from 2 to 1
move 1 from 1 to 2
```

## Branje podatkov

Naloga je trivialna, obupno zoprno pa je branje podatkov. Po mnogo poskusih biti pameten in elegantnen, sem končal pri tem. Ni lepo, je pa vsaj pregledno.

```
from copy import deepcopy

lines, instructions = open("example.txt").read().split("\n\n")
lines = lines.splitlines()
stacks = {lines[-1][column]: [lines[row][column]
                              for row in range(len(lines) - 2, -1, -1)
                              if len(lines[row]) > column and lines[row][column] != " "]
          for column in range(1, len(lines[-1]), 4)}

stacks2 = deepcopy(stacks)
```

Celotno vsebino datoteke razkosamo na `lines` in `instructions`.

Potem imamo zanko, ki šteje stolpce: `column` bo šel po x-koordinatah slike: prvi stolpec je na indeksu (znotraj vrstica) 1, drugi na 5 in tako naprej.

Zanka sestavlja slovar.

- Ključi so številke, ki jih preberemo v zadnji vrstici v tem stolpcu (`lines[-1][column]`). To bodo v resnici 1, 2, 3, 4, lahko pa bi bile tudi drugačne, ali pa bi imeli stolpci celo črkovne oznake. Številk ne spreminjamo v `int`-e, ker ni potrebe.
- Pripadajoče vrednosti so seznam paketov v tem stolpcu. Tega dobimo tako, da gremo z zanko po številkah vrstic in sicer od spodaj navzgor (ker

želimo, da so zgornji paketi na koncu seznama). V seznam zlagamo, kar je v tej vrstici in tem stolpcu (`lines[row][column]`), vendar le, če je ta vrstica tako dolga, da sploh pokriva ta stolpec in če je na tem mestu paket in ne presledek.

To je to. Grdo, ampak dela.

Na koncu naredimo kopijo vsega, za drugi del. Če bi naivno napisali `stacks2 = stacks.copy()`, bi dobili kopijo slovarja, ki pa bi vseboval *iste* elemente. Ker želimo, da so tudi elementi kopije, uporabimo funkcijo `deepcopy` iz modula `copy`.

Zdaj pa bomo kar vzporedno rešili oba dela.

Beremo navodila. Če jih razbijemo s `split`, bomo vedno dobili šest reči, pri čemer nas zanimajo, druga, četrta in šesta, ki povedo koliko paketov premikamo, odkod in kam.

Vzamemo torej navodila; s `strip()` odsekamo zadnji `\n`, da ne bo sitnosti; jih s `splitlines` razbijemo na vrstice, in nato vsak niz razbijemo s `str.split` na besede. Iz teh razbitij vzamemo `quantity`, `source` in `destination`.

Potem prelagamo tovor: v prvem delu prestavimo zadnjih `quantity` elementov v obratnem vrstnem redu, za drugi del v takšnem, kot je.

```
for _, quantity, _, source, _, destination in map(str.split, instructions.strip().splitlines
    quantity = int(quantity)

    # Part 1
    stacks[destination] += stacks[source][-1:-quantity-1:-1]
    del stacks[source][-quantity:]

    # Part 2
    stacks2[destination] += stacks2[source][-quantity:]
    del stacks2[source][-quantity:]
```

Na koncu izpišemo, kot zahteva naloga, imena vrhnjih paketov.

```
print("".join(stack[-1] for stack in stacks.values()))
print("".join(stack[-1] for stack in stacks2.values()))
```

```
CMZ
MCD
```